


Convolutional neural network for smoke and fire semantic segmentation

Sebastien Frizzi¹  | Moez Bouchouicha¹ | Jean-Marc Ginoux¹ | Eric Moreau¹ | Mounir Sayadi²

¹ Aix Marseille Univ, Université de Toulon, CNRS, LIS, Toulon, France

² University of Tunis, ENSIT, LR13ES03 SIME, Montfleury, Tunis 1008, Tunisia

Correspondence

Sebastien Frizzi, Aix Marseille Univ, Université de Toulon, CNRS, LIS, Toulon, France.
Email: sebastien.frizzi@univ-tln.fr

Funding information

PHC Utique program; French Ministry of Foreign Affairs and Ministry of Higher Education; Research and Innovation; Tunisian Ministry of Higher Education and Scientific Research, Grant/Award Number: 19G1126

Abstract

In recent decades, global warming has contributed to an increase in the number and intensity of wildfires destroying millions hectares of forest areas and causing many casualties each year. Firemen must therefore have the most effective means to prevent any wildfire from breaking out and to fight the blaze before being unable to contain and extinguish it. This article will present a new network architecture based on Convolutional Neural Network to detect and locate smoke and fire. This network generates fire and smoke masks in an RGB image by segmentation. The purpose of this work is to help firemen in assessing the extent of fire or monitor an incipient fire in real time with a camera embedded in a vehicle. To train this network, a database with the corresponding images and masks has been created. Such a database will allow to compare the performances of different networks. A comparison of this network with the best segmentation networks such as U-Net and Yuan networks has highlighted its efficiency in terms of location accuracy, reduction of false positive classifications such as clouds or haze. This architecture is also efficient in real time.

1 | INTRODUCTION

Each year, the news highlights the importance of fire detection when it comes to saving lives, wild forests and homes. Video images are able to detect and locate smoke and fire in real time and help firemen to act quickly. Therefore, most of the time, smoke is the first sign of a fire outbreak. Smoke detection and localization provide information such as starting points, size, type etc. It is essential to allow the firemen to organize the action plan to protect the population and the operation to put out the fire as quickly as possible. In the event of a wildfire, responsiveness is a very important factor in saving lives and protecting nature.

Yann Le Cun pointed out the use of Convolutional Neural Network (CNN) for classification in image learning. This type of neural network by their accuracy uninterrupted have kept growing for two decades. The substantial rate of improvements of this type of architecture has kept increasing for image classification [1, 2]. CNN's enhancements not only relate to the classification of images but also to the location of objects whose

bounding box methods are examples [3–5]. Kaiming He et al combine bounding box and segmentation to improve the object localization [6].

In recent years, semantic segmentation methods have been proposed using convolutions and deconvolutions architectures [7]. The main advantage of semantic segmentation of RGB images is to detect and locate objects in a single operation promptly and accurately. Generally, the network is trained by supervised learning based on examples input RGB images and output masks pairs.

We suggest studying and comparing different convolutional–deconvolutional architectures of neural network segmentation to detect and locate smoke and fire in RGB frames. Our goal is to find the best structure to segment smoke and fire compatible with real time.

Inspired by the success of fully convolutional network segmentation, we introduce in this article a new architecture based on the VGG16 [8] for the convolution phase. To increase the depth of our network and the size of the receptive field, we have replaced the fully connected layer of the VGG16

TABLE 1 Comparison between VGG16 and our network for the coding phase. Nb FM: number of feature maps. Size FM: feature map size. Feature map size for an example of a 640x480 RGB image

Coding phase					
VGG16 network		Our network			
Operations	Nb FM	Operations	Nb FM	Name	Size FM
Convolution + ReLu 3x3	64	Convolution + ReLu 3x3	64	FM1	640x480
Convolution 3x3 + ReLu	64	Convolution 3x3 + ReLu	64	FM2	640x480
MaxPooling 2x2	64	MaxPooling 2x2	64	FM3	320x240
Convolution 3x3 + ReLu	128	Convolution 3x3 + ReLu	128	FM4	320x240
Convolution 3x3 + ReLu	128	Convolution 3x3 + ReLu	128	FM5	320x240
MaxPooling 2x2	128	MaxPooling 2x2	128	FM6	160x120
Convolution 3x3 + ReLu	256	Convolution 3x3 + ReLu	256	FM7	160x120
Convolution 3x3 + ReLu	256	Convolution 3x3 + ReLu	256	FM8	160x120
Convolution 3x3 + ReLu	256	Convolution 3x3 + ReLu	256	FM9	160x120
MaxPooling 2x2	256	MaxPooling 2x2	256	FM10	80x60
Convolution 3x3 + ReLu	512	Convolution 3x3 + ReLu	512	FM11	80x60
Convolution 3x3 + ReLu	512	Convolution 3x3 + ReLu	512	FM12	80x60
Convolution 3x3 + ReLu	512	Convolution 3x3 + ReLu	512	FM13	80x60
MaxPooling	512	MaxPooling	512	FM14	40x30
Convolution 3x3 + ReLu	512	Convolution 3x3 + ReLu	512	FM15	40x30
Convolution 3x3 + ReLu	512	Convolution 3x3 + ReLu	512	FM16	40x30
Convolution 3x3 + ReLu	512	Convolution 3x3 + ReLu	512	FM17	40x30
MaxPooling 2x2	512	MaxPooling 2x2	512	FM18	20x15
Fully connected layer + ReLu	4096	Convolution 7x7	1024	Output coding phase	20x15
Fully connected layer + ReLu	4096				
Fully connected layer + Softmax	1000				

structure for a 7×7 convolution operation kernel Table 1. Removing fully connected layers frees us from the input size of images. For the decoding phase, we have chosen to use only three transposed convolutions to reach for the output masks the size of the input data. The output of the first and second up-sampling operation are combined with feature maps of the coding path [9] and followed by a convolution operation. Context information is propagated to the higher resolution layers by this sharing of the feature maps in the decoding path.

The paper is organized as follows: In Section 2, first of all, we have reviewed related work to convolutional neural network applied to semantic segmentation as well as the evolution of smoke and fire detection techniques. Then, in the same section, we describe our distinctive network architecture, the composition of our smoke/fire database and the evaluation parameters chosen to compare our network to Yuan [10] and U-Net [11] networks. The experimental results and discussion of our study are presented in the Section 3. Finally, the last section summarizes our work and lists the ways to improve semantic segmentation of smoke and fire.

2 | RELATED WORK

2.1 | Convolution neural network for semantic segmentation

Historically, first methods for fire and smoke detection in an image or video rely exclusively on colours. The latter have given satisfactory results. Some interesting works can be mentioned: Toreyin et al. did an extensive work on this field, in [12–15]. In [12], as an initial step in his fire and flame detection system, he used a hybrid background estimation for moving region detection. Afterwards, colours of moving pixels are compared with a colour distribution obtained from sample images containing fire regions. In the third step, he uses a temporal wavelet analysis to determine high activity region within these moving regions. Finally, he processed a spatial wavelet analysis of moving regions containing fire mask pixels to capture colour variations in pixel values. These two last steps are crucial in Toreyin's approach because of the turbulent high frequency behaviors on the boundary and inside a fire region. In [13] and [14], he enhanced his model by using separate Markov models for flame and non-flame moving pixels. He also, carried

out a flicker analysis by using HMMs and a wavelet domain analysis of object contours. Finally in [15], he updates his work using the least-mean-square (LMS) algorithm to combine the decisions from four sub-algorithms: (i) detection of fire coloured moving objects, (ii) temporal and (iii) spatial wavelet analysis for flicker detection, and (iv) contour analysis of flame boundaries. Similarly, Celik made a significant contribution in this area: [16–18]: The main originality in his work was the use of YCbCr colour space instead of the RGB one to construct a generic chrominance model for flame pixel classification. Moreover, he developed new rules in YCbCr colour space to alleviate the harmful effects of changing illumination and improved detection performance. Other methods similar and derived from those presented above can be found in [19].

These methods have brought an advanced solution to the field of fire and smoke detection in videos and images. Unfortunately, they remain sensitive to the problem of false alarms. Moreover, these methods require an expert to set the rules and features of the process for the object classification pipeline. On the other hand, there are methods based on neural networks that make it possible to overcome these weaknesses. In 2012, Alex K.'s work highlighted this type of method [1]: the so called "Deep Learning." This field predates Alex's work, and was initiated by Hinton [20], Lecun [21], Bengio [22] etc.

For a few years now, deep learning has become an essential tool for detecting smoke and fire in images or videos. This is due to the robustness of the algorithms used and the increasing availability of data. Sebastien [23] is one of the first researchers who used Convolutional Neural Network (CNN) to detect fire and smoke in a video stream. The CNN model inspired by AlexNet [1] operates directly on raw RGB frame without the need of the feature extraction stage. The CNN automatically learns a set of visual features from the training data. A classification accuracy score of 97 % was achieved. Similarly, Muhammad [24] used a fine-tuned CNN derived from Squeeze Net [25]. The latter allows the detection, localization and semantic interpretation of the fire scene to be carried out at the same time. More recently, Kim [26], proposed a net based on Faster Region-based Convolutional Neural Network (R-CNN) [27] to detect the suspected regions of fire (SRoFs) and of non-fire based on their spatial features. He also used Long Short-Term Memory (LSTM) [28] to interpret the dynamic fire behavior. The last methods give very good results, but that is still not enough, as the location of the fire or smoke region is not precise and is only characterized by a bounding box. To overcome this weakness, we have moved toward complete semantic segmentation. Indeed, semantic segmentation classifies all the pixels of the image, thus making the location of the fire or smoke very accurate.

Technical achievement of Convolutional Neural Network applied to semantic segmentation (CNN Segmentation) [9] has led us to apply this type of architecture to detect and locate smoke and fire. Smoke and fire are difficult to segment due to their non-constant shape and colour characteristics. The fire seems to be easier to segment than smoke due to its hues, but fire is less present at the start of a wildfire and de facto less

present in database images involving a difficulty in classifying it.

The U-net network architecture [11] is composed of an encoder-decoder with the distinctive particularity of sharing features maps from the convolution phase to the deconvolution phase.

Feiniu Yuan et al. [10] propose a smoke segmentation using CNN with an architecture composed of two different paths merging at the end to create the smoke mask. Both coding part are based on VGG16 architecture [8]. The coding path is followed by a successive up-sampling operations with concatenations of coding feature maps. The first path, which is deeper, provides global contextual information for smoke segmentation. The second shallower gives rich local information for smoke localization and object details.

2.2 | Our architecture

We assume a camera onboard a drone or a helicopter to locate a fire or smoke. Vehicle movements might not allow us to fix same spatial pixels in successive frames. This can prevent us from focusing on the temporal dynamic texture of fire and smoke. Our CNN architecture segments fire and smoke in each video frame without taking into account the temporal history of the pixels.

Our network (Figure 1) is based on VGG16 architecture [8] for coding phase. VGG16 is an architecture model proposed by K. Simonyan and A. Zisserman from the University of Oxford used for large-scale image recognition with good accuracy. We have chosen this structure for the coding phase due to the performances of features extraction for a large diversity of objects classification. VGG16 is composed of 13 convolution operation blocks with kernel 3×3 followed by three fully connected layers allowing object classification. For the coding phase, we kept from the VGG16 architecture for our network the five convolution blocks with a 3×3 kernel followed by a maxpooling operation. The dense layers of VGG16 structure set the input size of the image at 224×224 pixels. We have chosen to replace the fully connected layer by a convolutional operation with a 7×7 kernel giving 1024 feature maps. This approach keeps out from the issue of input images size (Table 1). We test different sizes for the last kernel (1×1 , 3×3 , 5×5 , 7×7 , 9×9). The 7×7 kernel size was the best compromise between accuracy segmentation and time consuming.

The purpose of the coding phase is to extract local information relating to fire and smoke. The deeper layers lose detail localization but increase the generalization capacity of the classification process. The decoding phase aims to recreate a high resolution segmentation of fire and smoke with good generalization. To achieve this objective, like U-Net network, we concatenate feature maps of the coding phase with the decoding phase to propagate contextual information to higher layers.

The decoding phase is composed of two transpose convolutions (up-sampling operation) with a 4×4 kernel and a last transpose convolutions with a 16×16 kernel Table 2. The wide

TABLE 2 Our network decoding phase. Nb FM: number of features maps. Size FM: feature map size. Feature map size for an example of a 640x480 RGB image

Decoding phase		
Operation type	Nb FM	Size FM
Output coding phase	1024	20x15
Deconvolution 4x4 Kernel	512	40x30
Concatenation with FM14	1024	40x30
Convolution 3x3 kernel + ReLU	512	40x30
Deconvolution 4x4 Kernel	256	80x60
Concatenation with FM10	512	80x60
Convolution 3x3 kernel + ReLU	256	80x60
Deconvolution 16x16 Kernel	3	640x480

“receptive field” of the transpose convolution kernel aims to increase the generalization capacity of the mask constructions. The first and second up-sampling operations are followed by a concatenation operation with the feature maps of coding phase and followed by convolution operation with a kernel 3X3. All convolution operations are followed by Rectified Linear Units ReLU activation function. The training parameter number of our network architecture is 57 million.

While U-Net architecture use four up-convolutions and Yuan eight with kernel 2X2, we use only three with the kernels 4X4, 4X4, and 16X16, respectively. Our coding path based on VGG16 is different from U-Net one. Our network differs from Yuan’s network due to a unique coding–decoding path, as well as the size of the kernel 7X7 of the last convolution of the coding phase.

2.3 | Our database

Database quality is of paramount importance to train deep network with good accuracy. We use internet images with different sizes and qualities. The presence in the database of different type of rather whitish or blackish smoke is also important to detect and segment correctly most types of the fires. We segmented 366 images and labelled them manually with Labelme software under Linux [29]. We performed offline data augmentation by flipping, cropping, rotating, adding noises, changing contrast/brightness and a combination of these transformations to reach 8784 images (Figures 2 and 3).

The 8784 images are divided into 82% to train our network (7224 images) and 18% (1560 images) to validate it. The validation images set is only used to follow the IoU (Intersection over union) metric for each class and avoid over-fitting. The weight and bias of the network are set during the validation phase.

2.4 | Evaluation parameters

We used the Python library Tensorflow 1.12.0 and Opencv 3.4.0 under Linux 18.04 to train our network and rise up the num-

ber of image data. We worked with GPU of a Nvidia GeForce 1080 graphic card with 11GB RAM. We initialized the parameters of the coder part of the network (weight of the first 13 convolution operations) by using a VGG16 pre-trained model on the ImageNet database. We trained our model on our train dataset with an Adam optimizer method [30] with a set learning rate of 5×10^{-5} and a cross entropy with logit loss function.

We compared our architecture with the U-Net [11] and Yuan [10] networks. To measure fairly the respective performances of these networks, we trained the three networks on our dataset. Unfortunately, we were unable to test our network with Yuan team dataset due to the absence of fire masks.

Training parameters for the U-Net and Yuan network follow the procedure explained in their research articles.

2.4.1 | Standard accuracy metrics

This sections describes metrics criteria used to compare the performances of segmentation for the different networks [31]. The confusion matrix (Figure 4) allows for each class and on all the valid images to calculate standard metrics to evaluate the performance of pixel classification.

Accuracy is a good tool to report the percentage of the correctly classified pixels in the image. We have chosen to report accuracy for each class. We calculated the average accuracy (1) on the N validation images for each class c. TP_i , TN_i , FP_i and FN_i are for the i th image, respectively, the true positives, true negatives, false positives and false negatives.

$$\overline{Accuracy} = \sum_{i=1}^N \left(\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \right). \quad (1)$$

Precision (2) provides a class agreement of the data labels with the positive labels given by the classifier,

$$\overline{Precision} = \sum_{i=1}^N \left(\frac{TP_i}{TP_i + FP_i} \right) = \sum_{i=1}^N \left(\frac{TP_i}{PredPositives_i} \right). \quad (2)$$

Recall (3) permit to assess the effectiveness of the network to identify positives labels with respect to the ground truth labels.

$$\overline{Recall} = \sum_{i=1}^N \left(\frac{TP_i}{TP_i + FN_i} \right) = \sum_{i=1}^N \left(\frac{TP_i}{TruePositives_i} \right). \quad (3)$$

We calculated metrics on valid images for each class and not global metrics because they are not appropriate when the representative frequency of the classes is unbalanced. Our database is unbalanced, the pixels of the smoke class are more frequent than those of the fire class.

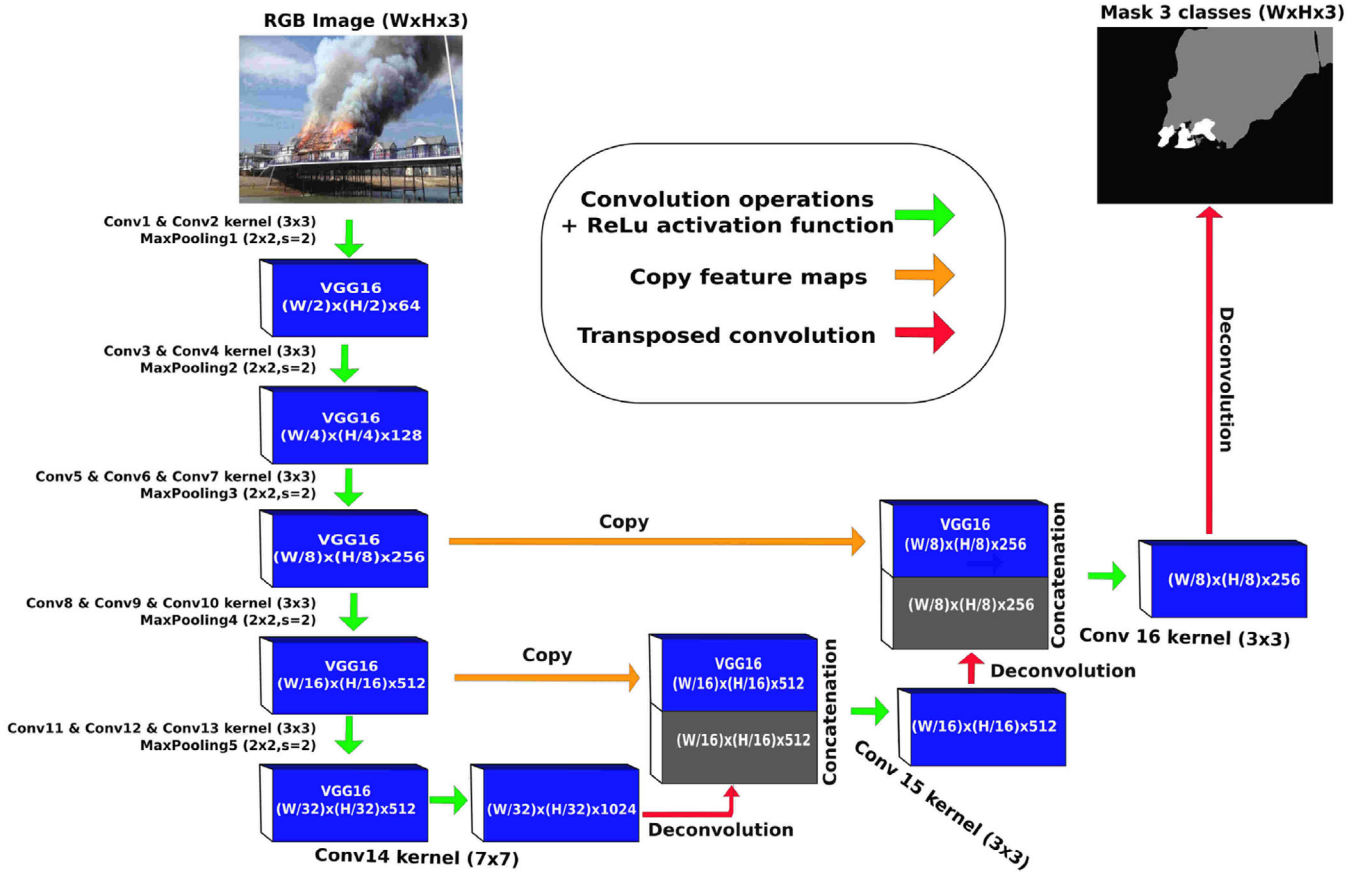


FIGURE 1 Network architecture

2.4.2 | Intersection over Union

Jaccard index or Intersection over Union (IoU) criterion (4) allows a quantitative evaluation of the accuracy segmentation. We used this criterion on the valid dataset by calculating the average IoU for each class (GroundTruth, smoke and fire)

$$\overline{IoU} = \sum_{i=1}^N \left(\frac{TP_i}{TP_i + FN_i + FP_i} \right). \quad (4)$$

2.4.3 | ROC curves

Receive Operating Characteristic (ROC) curve [32] (Figure 5) is a graphical representation of a model performance in function of the classification threshold. We used a Softmax function on the last feature maps to evaluate the likelihood of each pixel in order to verify if it belong to a given class or not. In addition to the area under the ROC curves [33], this evaluation method determines the behavior toward the false negatives or false positives of the model.

Finally, we selected two methods to define the optimal threshold which gives the maximum correct pixel classification

(Figure 5). The first consists in finding the optimal classification threshold by minimizing the distance d between the point (FPR=0,TPR=1) and the point(FPR,TPR) for a given threshold. The second method is based on maximizing the Youden index J [34] that maximizes the distance between the random chance line and the point (FPR,TPR) for a given threshold. Maximal J criterion is commonly used because it gives the threshold which maximizes the TPR and minimizes the FPR [35].

2.4.4 | Other criterion

We chose to plot the accuracy and IoU versus the threshold to evaluate the probability distribution of the pixel classification for a given class. We use the Softmax function at the output of the networks to calculate the probability of pixel prediction.

In addition, by observing the shape of the accuracy or IoU versus time, we could compare the ability of networks to segment classes. The decrease in the accuracy curves versus threshold (for high threshold) indicates a low proportion of high probability that the pixels belong to class c means lower segmentation performance.

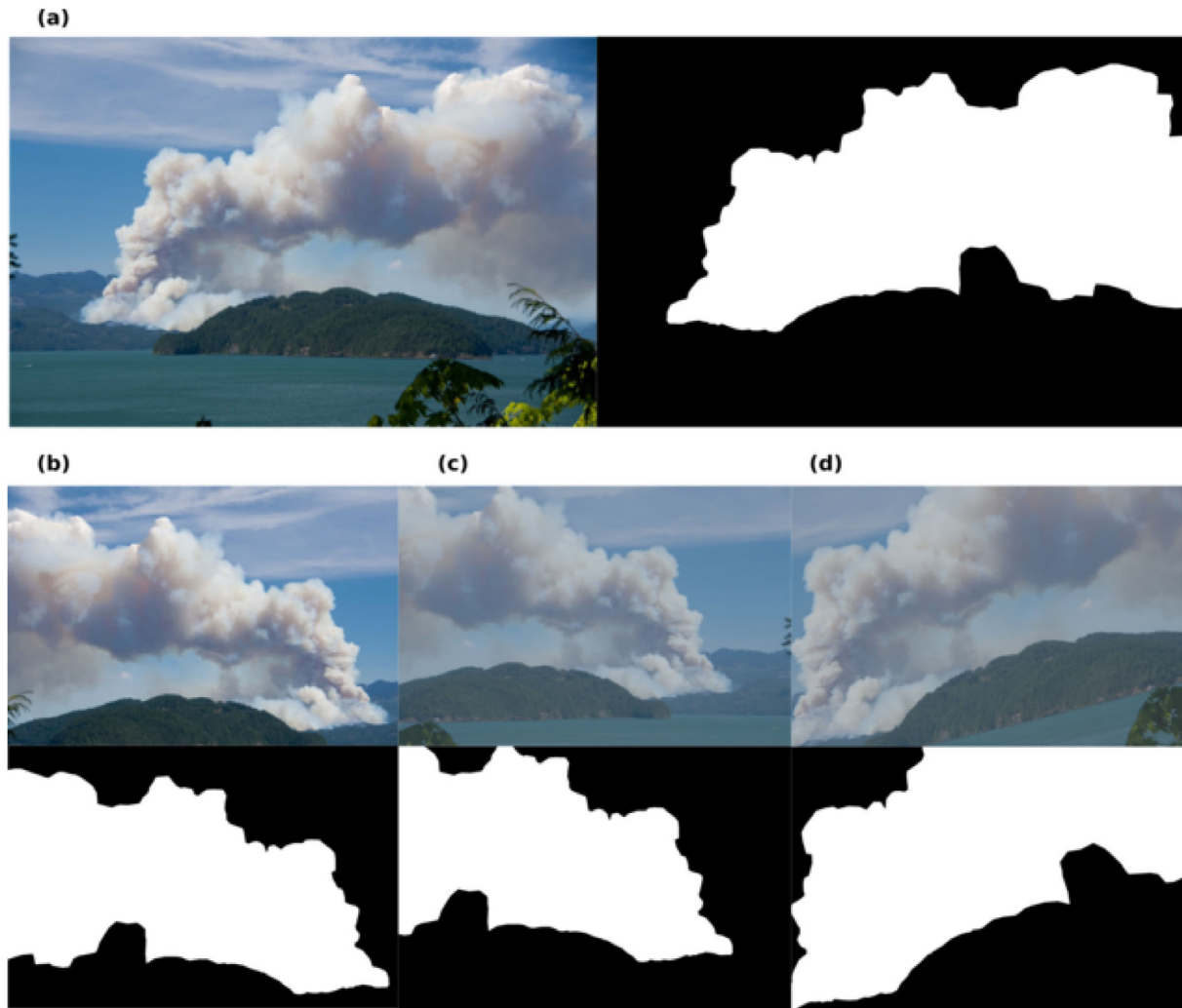


FIGURE 2 Example of augmentation image smoke: (a) original image and smoke mask; (b) flip and crop transformations; (c) flip and contrast variation transformations; (d) rotation and contrast variation transformations

TABLE 3 Average accuracy, precision, recall and IoU for background class

Background	Accuracy	Precision	Recall	IoU
Our network	0.934	0.916	0.939	0.864
U-Net network	0.902	0.872	0.915	0.806
Yuan network	0.924	0.899	0.934	0.846

TABLE 4 Average accuracy, precision, recall and IoU for Smoke class

	Accuracy	Precision	Recall	IoU
Our network	0.925	0.941	0.907	0.858
U-Net network	0.893	0.915	0.866	0.801
Yuan network	0.916	0.934	0.895	0.841

TABLE 5 Average accuracy, precision, recall and IoU for Fire class

	Accuracy	Precision	Recall	IoU
Our network	0.981	0.794	0.890	0.723
U-Net network	0.977	0.764	0.833	0.663
Yuan network	0.981	0.813	0.860	0.718

3 | EXPERIMENTAL RESULTS

In this section, we compare segmentation classification performance for smoke, fire and background in RGB images with different networks. We have chosen the last two best architectures for images segmentation which are U-Net network [11] and Yuan et al. network [10]. We have used the same validation images not yet seen by the network to compare network performances.

Tables 3–5 show that the U-Net network achieved the lowest performance in the background, fire and smoke pixels classifi-

cation. U-Net has low segmentation efficiency considering the IoU for each classes. The network of Yuan et al. achieves fire classification fairly well by equalizing the average accuracy with respect to our network. Nevertheless, the fire recall parameter

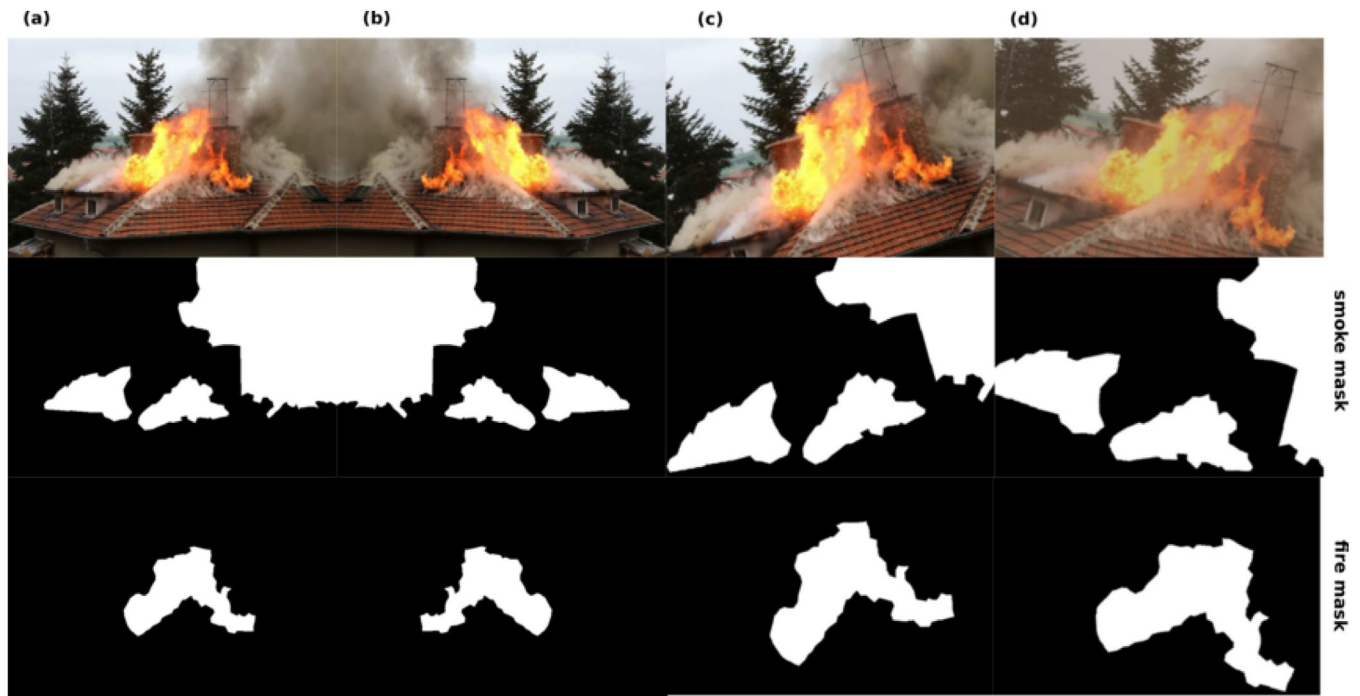


FIGURE 3 Example of augmentation image smoke and fire: (a) original image, smoke and fire mask; (b) flip transformation; (c) rotation and crop transformation; (d) rotation, crop and contrast variation transformation

		Predicted class		
		Positive	Negative	
True class	Positive	<i>TP</i>	<i>FN</i>	True positives
	Negative	<i>FP</i>	<i>TN</i>	True negatives
Total		Pred positives	Pred negatives	

FIGURE 4 Confusion matrix and notation

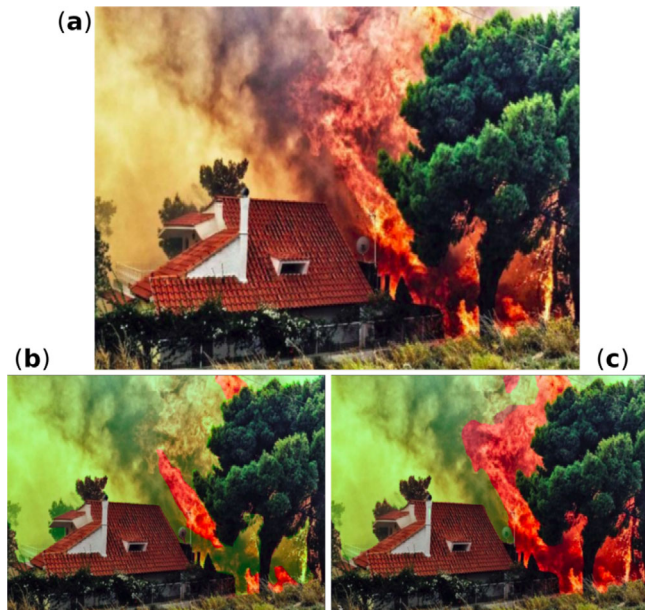


FIGURE 5 (a) RGB image. Area of smoke and fire are ambiguous (b) Ground truth segmentation (green: smoke and red: fire), you can note the smoke area segmented under the tree. (c) the predicted segmentation. The network segment fire area under the tree

for our network surpasses the Yuan network indicating a better classification. The IoU metrics prove the pre-eminence of our network to generalize segmentation over fire and smoke classes.

#ipr212046-tbl-0004.tab Fire IoU is lower than the smoke and the background classes for all the networks. The first explanation of this low value is due to the manual segmentation of the ground truth in our database. It seems easier to segment fire according to its distinctive red or orange colour but it does not. When we segment an image containing fire and smoke, it is difficult to separate the bounds between fire and smoke. Sometimes, we can see the fire behind the smoke. In this case, do we classify these areas as fire or smoke? The network sometimes detects fire where we had segment a smoke because it finds areas related to fire characteristics (Figure 6). This segmentation is not really false but the misinterpretation of the network decreases the value of the intersection over union for the fire and smoke. The second explanation is due to the unbalanced number of pixels between the three classes. Fire is less present in images than smoke and background. Therefore, a fire segmentation error will have a greater effect due to the small amount of fire pixels on the database.

To improve the IoU of our unbalanced database, we trained our network with a weighted cross-entropy loss [38]. The three classes are weighted by $w_c = \text{median}_{fc} / \text{freq}(c)$ to create a more balanced version of our model. $f(c)$ is the total number of pixels of the class c divided by the total number of pixels of images where c is present and media_{fc} is the median frequency of the frequency of the class c (Table 6). The IoU results for weighted loss showed a very small increase for all metrics for the fire class and a very small decrease for all the metrics for the smoke

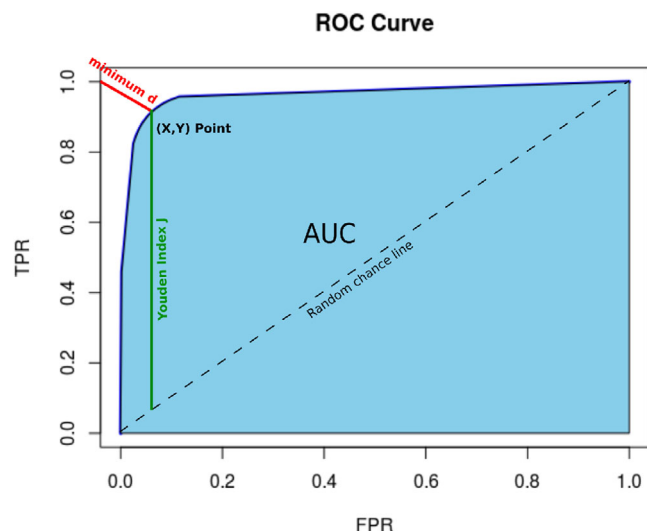


FIGURE 6 ROC curve and its components

TABLE 6 Weight for the three classes for the weighted cross-entropy loss

	Background	Smoke	Fire
$median_fc$	0.510	0.418	0.064
f_c	0.503	0.435	0.121
w_c	1.01	0.962	0.529

and background (Table 7). Smoke is the first fundamental information visible to detect a wildfire outbreak. The decrease in smoke metrics and the weak improvement in fire metrics incited us to maintain an unweighted loss function to train the networks.

Discrete ROC curves for each class are superior to the U-Net and Yang network. The areas under the curve for our model (Table 8) have the highest values that are close to the unit. That indicates the superiority of our prediction model for the three classes. Moreover, ROC curves of our network near the point of origin increase faster, pointing out a lower rate false positives classification whether it is fire, smoke or background classes.

TABLE 7 Comparison between weighted and non-weighted cross-entropy loss for the three classes. Metrics: Average accuracy, precision, recall and IoU

Background	Accuracy	Precision	Recall	IoU
Our network	0.934	0.916	0.939	0.864
Our network weighted	0.931	0.908	0.943	0.860
Smoke				
Our network	0.925	0.941	0.907	0.858
Our network weighted	0.923	0.942	0.901	0.854
Fire				
Our network	0.981	0.794	0.890	0.723
Our network weighted	0.983	0.819	0.890	0.744

TABLE 8 AUC for background, smoke and fire classes

AUC values	AUC background	AUC smoke	AUC fire
Our network	0.973	0.963	0.970
U-Net network	0.914	0.906	0.908
Yuan network	0.964	0.958	0.969

TABLE 9 Softmax average probabilities repartition of the true positives values for the smoke and fire

Network type	TP average probabilities	TP standard deviation probabilities
Smoke		
Our network	0.987	0.056
U-Net	0.571	0.012
Yuan et al.	0.762	0.068
Fire		
Our network	0.979	0.072
U-Net	0.570	0.020
Yuan et al.	0.757	0.079

The curve of the Houden index versus classification threshold provides information on the shape of the ROC curve. The faster the curve increases for the low threshold, the closer the ROC curve is to the perfect classification model. A value of the Houden index close to the unit also indicates a good classification. Houden index curve (Figures 7 and 8) highlights a long plateau for high value for our network, whether it be for smoke or fire, which means a high range of classification thresholds achieving an excellent segmentation with a maximum of true positives rate and a minimum of false positives rate. We have chosen not to draw the d measures because they are strongly correlated with the Houden index. U-Net and Yuan networks

Smoke – ROC Curve Our Network, U-Net and yuan

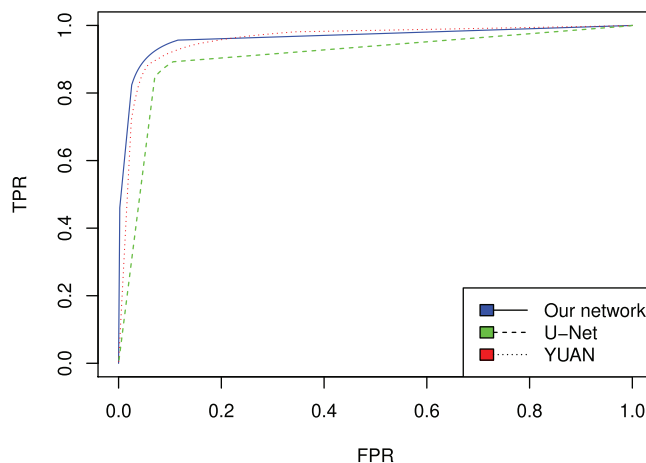
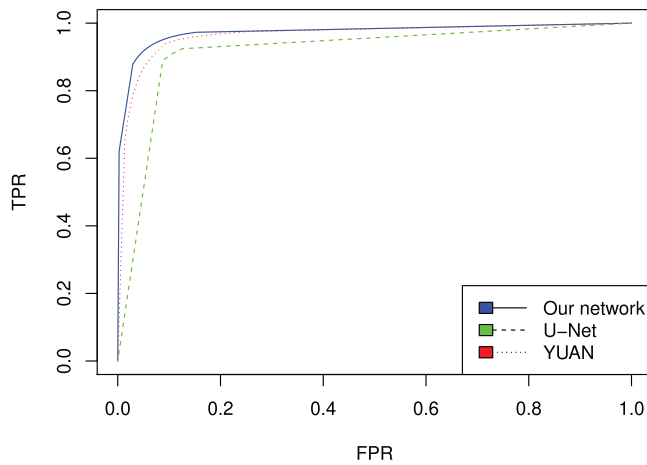
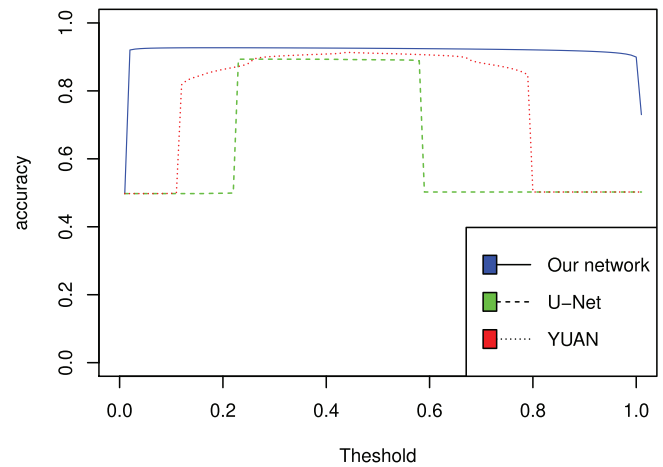
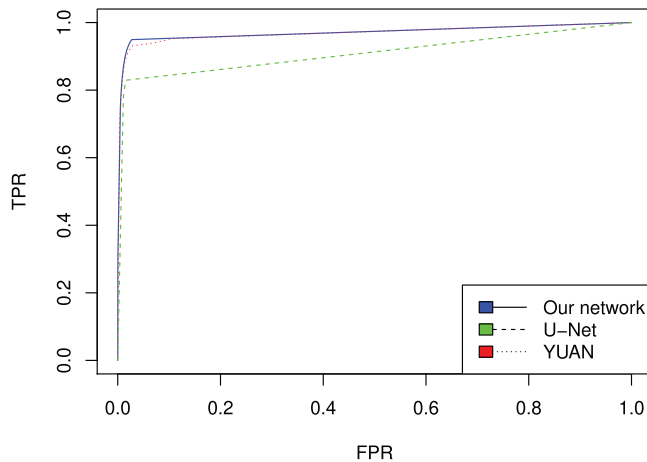
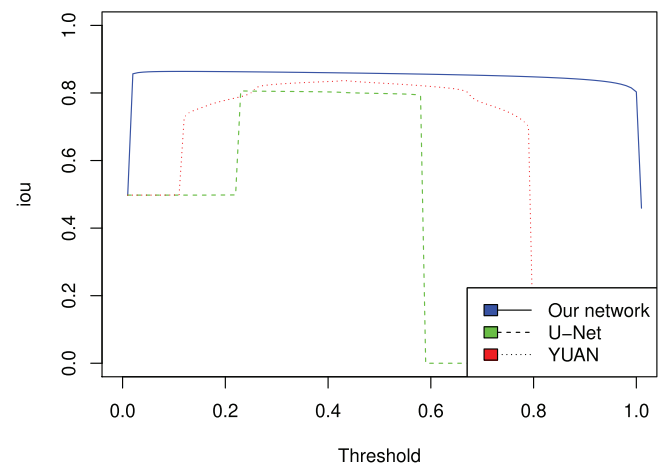


FIGURE 7 ROC curves for background class

Bgd – ROC Curve Our Network, U-Net and yuan**FIGURE 8** ROC curves for background class**accuracy Smoke****FIGURE 10** Accuracy versus threshold for smoke class**Fire – ROC Curve Our Network, U-Net and yuan****FIGURE 9** ROC curves for fire class**IoU smoke****FIGURE 11** IoU versus threshold for smoke class

possess a thinner band and a lower Houden index emphasizing a poorer fire and smoke segmentation performance than our network.

The accuracy and IoU curves are plotted according to the threshold which is directly related to the probability prediction of a pixel belonging to the class c . We use the Softmax function at the output of the networks to calculate this probability of pixel prediction.

The accuracy with respect to the threshold provides information on the percentage of the correctly classified pixels for a c class. A large plateau between a low threshold and a threshold close to the unit indicates that the majority of pixels in class c have a high accuracy for a high threshold and de facto a high prediction probabilities. Relating to smoke accuracy curves (Figure 9), we notice a large plateau between few percent threshold and 100% for our network compared with Yuan and U-Net network. The large and constant value of the plateau seems to indicate a clustering of high classification probabilities of the pixels

of the smoke class. These analyzes are correlated with the probability distributions of the true positives for the smoke and fire classes (Figure 10 and 12) (the pixel class is assigned to the highest Softmax probability class).

The same pattern as the accuracy curve one can be observed with the IoU curves (Figure 11 and 13). We can interpret this large plateau of the curve between a threshold of few percent and a threshold of 100% to a very high probability of classification of smoke pixel. For U-Net and Yuan network, the IoU curves decrease for, respectively, 60% and 80% indicating a drop of the locate accuracy segmentation for the high probabilities of smoke pixels classification. The same analysis can be done for the segmentation of the fire (Figures 14 and 15). Nevertheless, the drop of the curve has a lower impact than the smoke curve for the U-Net and Yuan network, which reveals a better pixel segmentation for the fire class than for the smoke class. The IoU and accuracy versus threshold curves assert, for our method, a

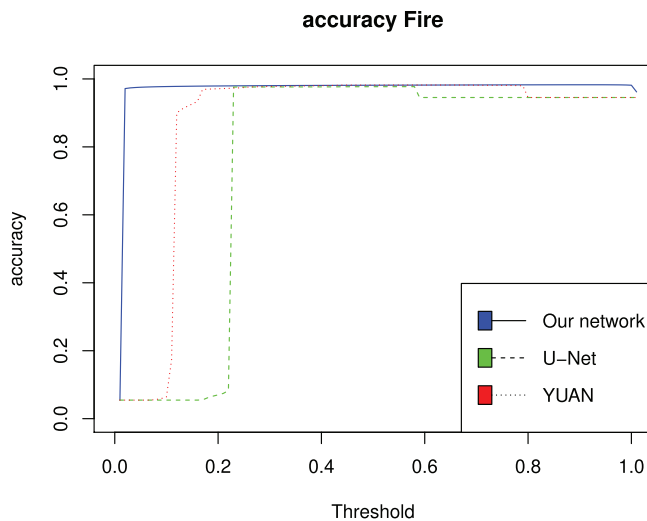


FIGURE 12 Accuracy versus threshold for fire class

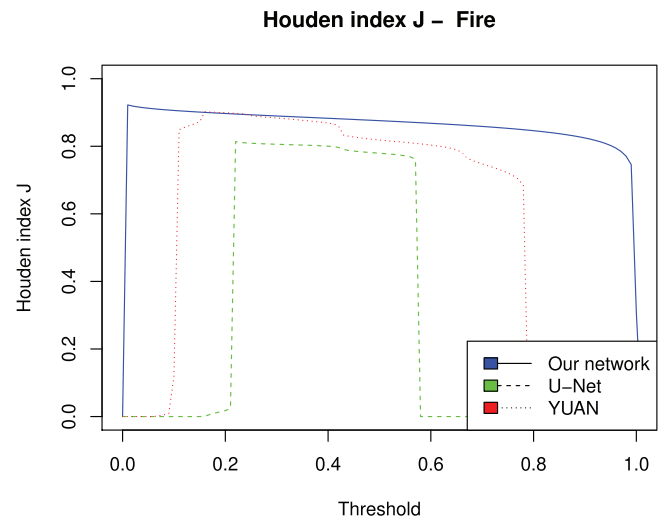


FIGURE 15 Houden index J versus threshold for fire class

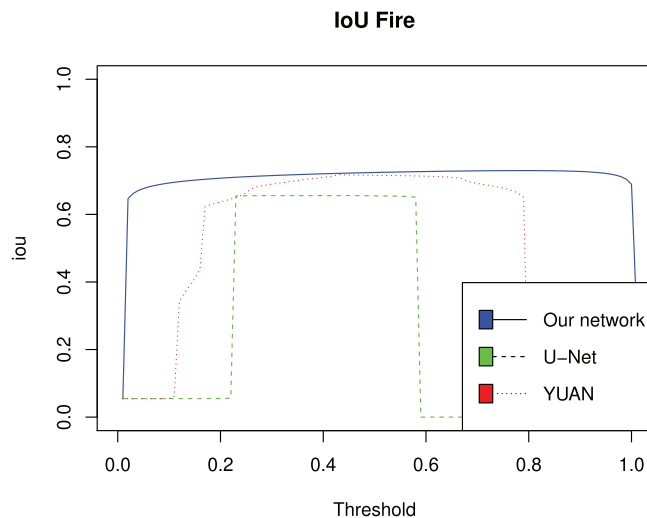


FIGURE 13 IoU versus threshold for fire class

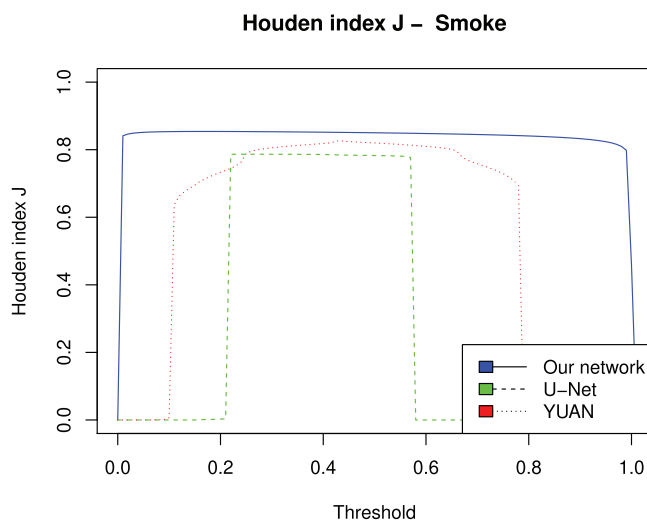


FIGURE 14 Houden index J versus threshold for smoke class

TABLE 10 Number of parameters of the network in millions and the time rate segmentation for images of 640×480 RGB with Nvidia GTX1080TI graphic card

Network type	Number of parameters (millions)	Segmentation rate time (image/s)
Our network	57.0	21.1
UNet	33.1	11.0
Yuan et al.	29.9	5.8

better segmentation of fire and smoke with less false positives and false negatives (Table 9).

#ipr212046-fig-0012.fig #ipr212046-fig-0013.fig The Table 10 compares the three network characteristics. Our network is the deepest with 57 million train parameters. However, our network is the fastest to segment images with the three classes due to the smaller number of up-sampling operations, the smaller number of high resolution features maps and only a single coding-decoding path. Our network is almost two times faster than U-Net network and almost four times faster than Yuan network.

#ipr212046-tbl-0009.tab Our architecture with a segmentation rate time greater than 20 frames per seconde is able to segment fire and smoke in a video 640x480 size in real time.

Figure 16 exhibits different images which clearly show the smoke mask predicted in green and the fire mask predicted in red for our network, U-Net and Yuan network.

Our network possesses the architecture with the lowest number of up-sampling operations (5+3 for Yuan, 4 for U-Net and 3 decoding transformation for our network). It can be assumed that the number of up-sampling operation is not an essential parameter for creating accurate smoke and fire segmentation.

The effective size of the receptive field is an important parameter in deep learning [36]. For a dense prediction such as segmentation image, it is essential for each pixel class of the output mask to have a large receptive field on the input image to

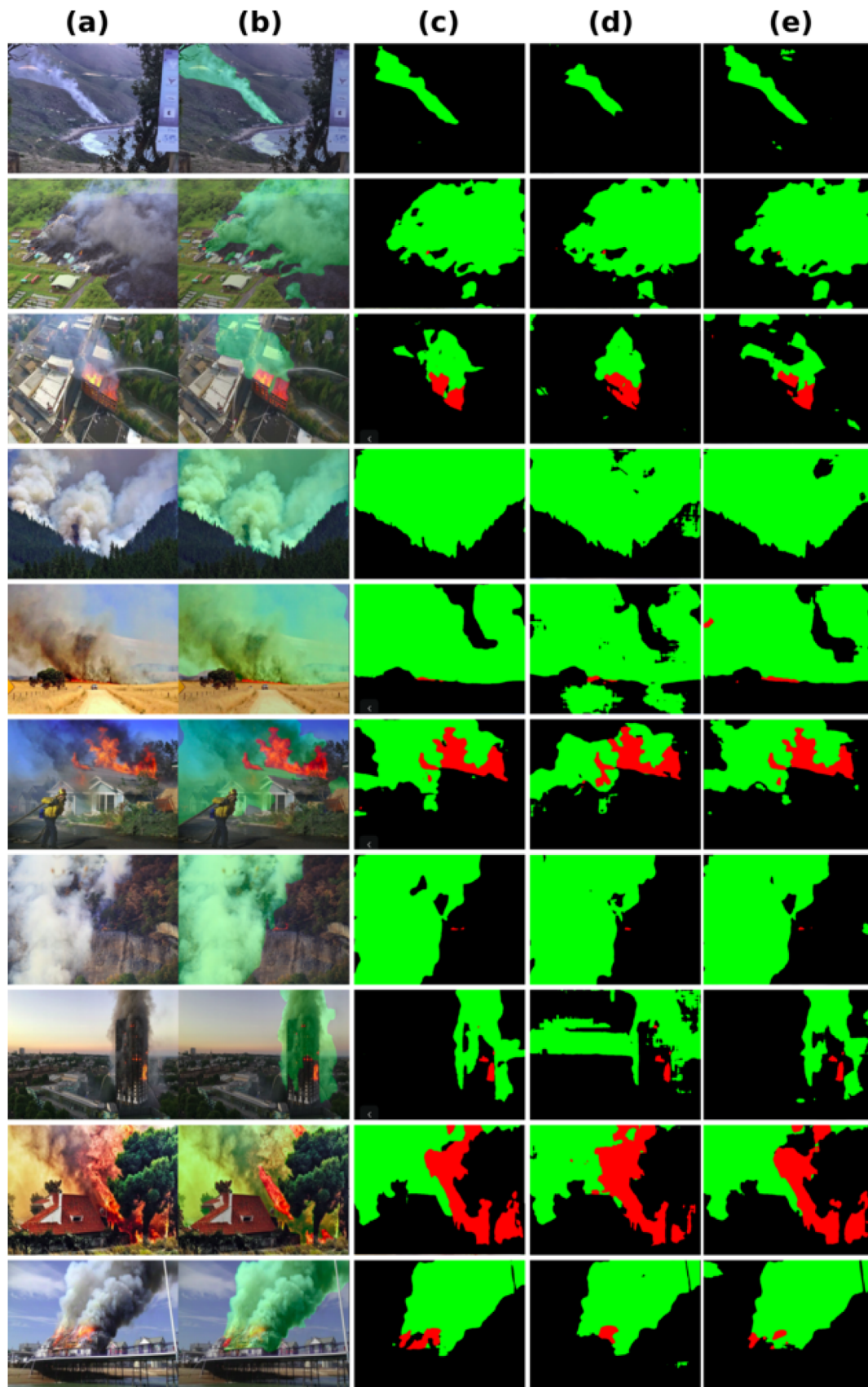


FIGURE 16 Mask prediction segmentation (a) original image. (b) Superimposition smoke mask in green, fire in red on original image. (c) Our network masks (green smoke and red fire). (d) U-Net network masks. (e) Yuan network masks

ensure that important information in the image is not omitted. Our network possesses the highest receptive field for the encoding phase due to the last 7×7 convolution operation (These effective receptive fields are respectively for our network, U-Net and Yuan: 404, 140 and 196. For our network, a pixel mask PM of coordinates (x,y) are influenced by information given by pixels of the input RGB images in a windows of 404×404 centred around PM position.)

Our generative mask method is close to the ground truth masks (e.g. Figure 16). False positives for fire and smoke classes are less prevalent with our network than in the Yuan and U-Net network. In addition, our method misclassified a small number of cloud pixels compared to the U-Net and Yuan methods. The quality of the segmentation can be explained by the large size of the receptive field and the depth of our network.

This article presents a new network architecture for segmenting smoke and fire in RGB images. We mainly compared our architecture with that of the Yuan. However, to prove that the good performances achieved by our network architecture is independent of the database we created, we decided to test it on the Yuan database.

We trained our network and Yuan's one on the Yuan database [39]. The latter is made up of 70,632 synthetic RGB images of size 256×256 pixels and their corresponding smoke masks. We split it into two sets: the train set (80%) and the valid set (20%). Yuan database contains three test datasets named DS01, DS02 and DS03. Each test set consists of 1000 256×256 RGB images and corresponding 8 bits ground truth of alpha channels. Using the ground truth of alpha channels, we created smoke masks (Figure 17). We had to choose a threshold to create smoke masks from the groundtruth of alpha channels because low values of the alpha channels for the smoke were not visible on the RGB images. We chose the value 20 for this threshold; that is, pixels of the alpha channels with values under 20 were considered as background and values over or equal to 20 were considered as smoke.

We tested performances of the networks by calculating the IoU (4) and mMse (5) the average square difference per pixel between the prediction and the ground truth on the test datasets (DS01, DS02 and DS03).

$$mMse = \frac{1}{N \times b \times w} \sum_{i=1}^N \sum_{k=1}^{b \times w} (Pred(x_k) - Gtruth(x_k))^2$$

$$\text{for 2 classes: } mMse = \frac{1}{N \times b \times w} \sum_{i=1}^N (FP_i + FN_i)$$

N is the number of images of the test set,

b and w are, respectively, the height and the width of images,

$pred(X_k)$ is the prediction of the pixel X_k ,

and $Gtruth(X_k)$ is the ground truth of the pixel X_k .

Tables 11 and 12 show that results of segmentation performances for both architecture on DS01 are almost similar. On

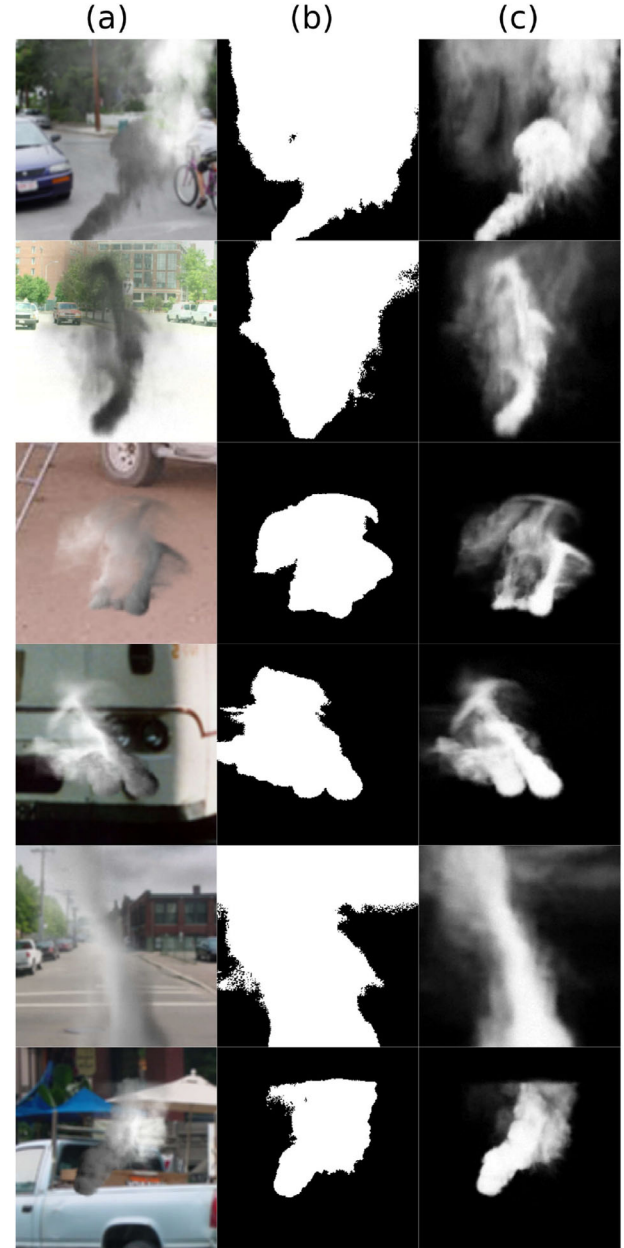


FIGURE 17 (a) Examples of Synthetic RGB images 256×256 pixels of the Yuan test datasets (DS01, DS02 and DS03). (b) Smoke masks generated with threshold 20. (c) Synthetic smoke: Ground truth of alpha channels

TABLE 11 IoU smoke segmentation on Yuan test datasets (4)

Network	IoU (%) DS01	IoU (%) DS02	IoU (%) DS03
Yuan network	70.45	69.03	69.47
Our network	70.43	70.08	70.70

TABLE 12 mMse smoke segmentation on Yuan test datasets (5)

Network	mMse DS01	mMse DS02	mMse DS03
Yuan network	0.110	0.120	0.116
Our network	0.109	0.115	0.110

TABLE 13 Average prediction execution time for a 256×256 px RGB image on the DS01, DS02 and DS03 test dataset

Network	Prediction time (mS)	FPS: frame per second
Yuan network	30.7	32.5
Our network	16.6	60.4

the other hand, results achieved on DS02 and DS03 datasets thanks to our network architecture outperform those of Yuan's one. Moreover, Table 13 indicates the execution time of the smoke mask prediction for a 256×256 px RGB image. Our network is twice as fast as the Yuan network. We can argue that in case of higher definition images, our architecture would still have results in real-time. This study has proven the quality of our network architecture for semantic segmentation compatible with real-time.

4 | CONCLUSION

Recently, full convolution networks have provided architectures to accurately segment objects in an image. Fire and smoke are objects with wide variety of shape and colours. Despite the difficulty of detecting and locating such objects, our network composed of a coding and decoding phase achieves a much better segmentation task than the Yuan and U-Net networks. Our method has demonstrated accuracy in classifying pixels with low false positives such as clouds or haze. Time consumed is also an important factor in segmenting fire and smoke according to real-time compatibility. Our network outperforms the other architectures for segmentation time.

To improve the segmentation accuracy of the fire class, we could increase the number of fire images in our database (we could add fire images coming from other database to our database like [37]). We could also, when the camera is almost static, use 3D convolutions to capture the dynamics of smoke and fire in successive frames of a video.

Our network outperforms U-Net and Yuan networks for the semantic segmentation method of smoke and fire in terms of location accuracy and segmentation rate time.

ACKNOWLEDGMENTS

We thank the Institute of Technology (IUT) of Toulon via the research committee CARTT and LIS laboratory for their financial and technical help.

ORCID

Sebastien Frizzi  <https://orcid.org/0000-0003-2576-961X>

REFERENCES

- Krizhevsky, A., et al.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* 60(6), 1097–1105 (2012)
- Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: *European Conference on Computer Vision*, pp. 818–833. Springer International Publishing, New York (2014)
- Redmon, J., et al.: You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788. IEEE, Piscataway, NJ (2016)
- Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271. IEEE, Piscataway, NJ (2017)
- Sermanet, P., et al.: Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:13126229*, (2013)
- He, K., et al.: Mask r-cnn. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2961–2969. IEEE, Piscataway, NJ (2017)
- Badrinarayanan, V., et al.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39(12), 2481–2495 (2017)
- Simonyan, K., Zisserman, A.: Preprint repository arxiv achieves milestone million uploads. *arXiv preprint*, (2014)
- Long, J., et al.: Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, (2015), pp. 3431–3440
- Yuan, F., et al.: Deep smoke segmentation. *Neurocomputing* 357, 248–260 (2019)
- Ronneberger, O., et al.: U-net: Convolutional networks for biomedical image segmentation. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241. Springer International Publishing, New York (2015)
- Töreyn, B.U., et al.: Computer vision based method for real-time fire and flame detection. *Pattern Recognit. Lett.* 27(1), 49–58 (2006)
- Töreyn, B.U., Cetin, A.: Online detection of fire in video. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–5. IEEE, Piscataway, NJ (2007)
- Töreyn, B.U., et al.: Hmm based falling person detection using both audio and video. In: *2006 IEEE 14th Signal Processing and Communications Applications*, pp. 1–5. IEEE, Piscataway, NJ (2006)
- Günay, O., et al.: Fire detection in video using LMS-based active learning. *Fire Ecol.* 46(3), 551–577 (2010)
- Celik, T., Demirel, H.: Fire detection in video sequences using a generic color model. *Fire Saf. J.* 44(2), 147–158 (2009)
- Celik, T., et al.: Fire detection using statistical color model in video sequences. *J. Visual Commun. Image Represent.* 18(2), 176–185 (2007)
- Celik, T., et al.: Fire and smoke detection without sensors: Image processing based approach. In: *2007 15th European Signal Processing Conference*, pp. 1794–1798. IEEE, Piscataway, NJ (2007)
- Çetin, A.E., et al.: Video fire detection—review. *Digital Signal Process.* 23(6), 1827–1843 (2013)
- Rumelhart, D.E., et al.: *Learning internal representations by error propagation*. La Jolla, Institute for Cognitive Science, University of California, San Diego (1985)
- Lecun, Y., et al.: Gradient-based learning applied to document recognition. *Proc. IEEE* 86(11), 2278–2324 (1998)
- Bengio, Y., et al.: Greedy layer-wise training of deep networks. In: *Advances in Neural Information Processing Systems 19*, pp. 153–160. The MIT Press, Cambridge (2007)
- Frizzi, S., et al.: Convolutional neural network for video fire and smoke detection. In: *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, pp. 877–882. IEEE, Piscataway, NJ (2016)
- Muhammad, K., et al.: Efficient deep cnn-based fire detection and localization in video surveillance applications. *IEEE Trans. Syst. Man Cybern., Syst.* 49(7), 1419–1434 (2018)
- Iandola, F.N., et al.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. *arXiv preprint*, (2016)
- Kim, B., Lee, J.: A video-based fire detection using deep learning models. *SN Appl. Sci.* 9(14), 2862 (2019)
- Ren, S., et al.: Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, pp. 91–99. Curran Associates, New York (2015)

28. Hochreiter, S., Schmidhuber, J.: LSTM can solve hard long time lag problems. In: *Advances in Neural Information Processing Systems, NIPS'9*, pp. 473–479. MIT Press, Cambridge (1997)
29. Russell, B.C., et al.: Labelme: A database and web-based tool for image annotation. *Int. J. Comput. Vision* 77 (1–3), 157–173 (2007)
30. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint, arXiv:1412.6980*, (2014)
31. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. *Information Processing & Management* 45(4), 427–437 (2009)
32. Egan, J.P.: *Signal Detection Theory and ROC-Analysis*. Academic Press, London (1975)
33. Hanley, J.A., Mcneil, B.J.: The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143(1), 29–36 (1982)
34. Youden, W.J.: Index for rating diagnostic tests. *Cancer* 3(1), 32–35 (1950)
35. Perkins, N.J., Schisterman, E.F.: The inconsistency of “optimal” cut-points obtained using two criteria based on the receiver operating characteristic curve. *American Journal of Epidemiology* 163(7), 670–675 (2006)
36. Luo, W., et al.: Understanding the effective receptive field in deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 4898–4906. Curran Associates, New York (2016)
37. Toulouse, T., et al.: Computer vision for wildfire research: an evolving image dataset for processing and analysis. *Fire Saf. J.* 92188–194 (2017)
38. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2650–2658. IEEE, Piscataway, New Jersey (2015)
39. Yuan, F.: Yuan smoke database (images and masks). <http://staff.ustc.edu.cn/yfn/>, (2019)

How to cite this article: Frizzi S, Bouchouicha M, Ginoux Jean-Marc, Moreau E, Sayadi M. Convolutional neural network for smoke and fire semantic segmentation. *IET Image Process.* 2021;15:634–647. <https://doi.org/10.1049/ipr2.12046>